

**Exercice 1** \exists et \forall

Q1 Écrire une fonction `contient1` prenant en argument une liste `L` et renvoyant `True` si cette liste contient le nombre 1 et `False` sinon. Par exemple, `contient1([2, 5, 1, 0, 1, 3])` doit renvoyer `True`.

Q2 Écrire une fonction `positif` prenant en argument une liste de nombres `L` et renvoyant `True` si tous les éléments de `L` sont positifs et `False` sinon. Par exemple, `positif([3, 6, -2, 1, 7])` doit renvoyer `False`.

Exercice 2 Selection artificielle

Q1 Écrire une fonction `tronque` prenant en argument une liste de nombres `L` et un nombre réel `m` et qui renvoie la liste des éléments de `L` plus petits que `m`. Par exemple, `tronque([4, 1, 7, 10, 5, 6, 7], 6)` doit renvoyer `[4, 1, 5, 6]`.

Q2 Écrire une fonction `select` prenant en argument une liste de nombres entiers `L` et renvoyant la liste formée des éléments de `L` qui sont divisibles par 3. Par exemple `select([1, 6, 5, 5, 3, 12])` doit renvoyer `[6, 3, 12]`.

Exercice 3 Dernières Nouvelles d'Alsace (début)

L'objectif de cet exercice est de manipuler les listes Python pour modéliser des brins d'ADN. Pour rappel, l'ADN est une macromolécule constituée de brins formés d'un enchaînement de nucléotides contenant les bases de l'ADN : Adénine (A), Thymine (T), Guanine (G) et Cytosyne (C).

Dans tout l'exercice, on représentera un brin d'ADN par une liste dont les éléments sont les chaînes de caractères `'A'`, `'T'`, `'C'` ou `'G'`. Par exemple, `['A', 'C', 'A', 'T', 'A', 'G']` représente un brin d'ADN formé de 6 bases.

Q1 Écrire une fonction `verif` prenant en argument une liste `L` et renvoyant `True` si cette liste peut correspondre à un brin d'ADN, et `False` sinon. Par exemple, `verif(['A', 'T', 'B'])` doit renvoyer `False`.

Une étape importante de l'utilisation de l'ADN consiste en sa transcription en ARNm. Il s'agit d'un autre brin, de même longueur et dont les bases sont : Adénine (A), Uracile (U), Guanine (G) et Cytosyne (C). La correspondance entre les bases de l'ADN et celles de l'ARNm est la suivante :

ADN	ARNm
Adénine (A)	Uracile (U)
Thymine (T)	Adénine (A)
Guanine (G)	Cytosyne (C)
Cytosyne (C)	Guanine (G)

Q2 Écrire une fonction `adn_to_arm` réalisant la transcription d'une séquence d'ADN en la séquence d'ARNm correspondante. Par exemple, si `L=['A', 'T', 'C', 'G', 'A', 'T']` alors `adn_to_arm(L)` doit renvoyer `['U', 'A', 'G', 'C', 'U', 'A']`.